



AF\$
JPL

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
Re: Appeal to the Board of Patent Appeals and Interferences

In re PATENT application of
ERIMLI

Group Art Unit: 2151

Application No. 10/046,784

Examiner: DIVECHA, Kamal B

Filed: January 16, 2002

Docket : 95-507

Title: ARRANGEMENT IN AN INFINIBAND CHANNEL ADAPTER FOR SHARING
MEMORY SPACE FOR WORK QUEUE ENTRIES USING MULTIPLY-LINKED LISTS

Date: March 27, 2006

Commissioner of Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

- 1 ☐ **NOTICE OF APPEAL:** Applicant hereby appeals to the Board of Patent Appeals and Interferences from the decision (not Advisory Action) dated October 26, 2005 of the Examiner finally rejecting claims 1-7
- 2 ☒ **BRIEF** on appeal in this application attached.
- 3 ☐ An **ORAL HEARING** is respectfully requested under Rule 194 (due two months after Examiner's Answer unextendable).
- 4 ☐ Reply Brief is attached (due two months after Examiner's Answer -- unextendable).

5. FEE CALCULATION:		Large/Small Entity	
If box 1 above is X'd, see box 12 below <u>first</u> and decide: enter		\$500/250*	\$
If box 2 above is X'd, see box 12 below <u>first</u> and decide: enter		\$500/250*	\$ 500.00
If box 3 above is X'd, see box 12 below <u>first</u> and decide: enter		\$1000/500*	\$
If box 4 above is X'd, enter nothing		- 0 - (no fee)	
6. Original due date: Monday, March 27, 2006			
7. Petition is hereby made to extend the original due date to cover the date this response is filed for which the requisite fee is attached	(1 mo) \$120 (2mos) \$450 (3mos) \$1020 (4mos) \$1590	+	
8. Enter any previous extension fee paid [] previously since above <u>original</u> due date (item 6); [] with concurrently filed amendment		-	
9. Subtract line8 from line7 and enter: Total Extension Fee			+\$0
10. TOTAL FEE ATTACHED =			\$ 500

11. ☐ *Fee **NOT** required if/since paid in prior appeal in which the Board of Patent Appeals and Interferences did not render a decision on the merits.

CHARGE STATEMENT: The Commissioner is hereby authorized to charge any fee specifically authorized hereafter, or any missing or insufficient fee(s) filed, or asserted to be filed, or which should have been filed herewith or concerning any paper filed hereafter, and which may be required under Rules 16-18 (missing or insufficient fee only) now or hereafter relative to this application and the resulting Official document under Rule 20, or credit any overpayment, to our Account/Order No. 50-0687 / 95-507 for which purpose a duplicate copy of this sheet is attached. This CHARGE STATEMENT does not authorize charge of the issue fee until/unless an issue fee transmittal form is filed

Atty: Leon R. Turkevich
Leon R. Turkevich
Reg. No. 34,035
Tel: (202) 261.1000

Customer No. 20736

Docket No.: 95-507



PATENT

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of

ERIMLI

Serial No.: 10/046,784

Filed: January 16, 2002

:
:
:
:
:
:

Group Art Unit: 2151

Examiner: DIVECHA, Kamal B

For: ARRANGEMENT IN AN INFINIBAND CHANNEL ADAPTER FOR SHARING
MEMORY SPACE FOR WORK QUEUE ENTRIES USING MULTIPLY-LINKED
LISTS

MAIL STOP: APPEAL BRIEF - PATENTS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

This is an appeal from the final rejection of claims 1-13 in the above-identified patent application.

This Appeal Brief is submitted as required by 37 C.F.R. §41.37.

1. Real Party in Interest:

This application is assigned to Advanced Micro Devices, Inc., the real party of interest.

2. Related Appeals and Interferences:

There are no other appeals or interferences known to Appellant that will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

03/28/2006 07:01:11 PM 00000030 10046784

01 FC:1402

500.00 0P

Appeal Brief filed Monday, March 27, 2006

Appln No. 10/046,784

Page 1

3. Status of Claims:

Claims 1-7 are pending in this application. Claims 1-7 stand rejected by the Examiner, and claims 1-7 are appealed.

4. Status of any Amendment File Subsequent to Final Rejection:

No Amendment was filed in response to the Final Rejection. A Response to the Final Rejection was filed on December 27, 2005.

5. Summary of Claimed Subject Matter:

The claimed subject matter includes independent claims 1 and 5 and dependent claims 2-4 and 6-7. Independent claims 1 and 6 each specify a channel adapter (12 of Fig. 1) that stores, in a table (20 of Fig. 2), entries (24 of Fig. 2) identifying transmitted work queue entries (e.g., "WQE1" of Fig. 2) in a work queue entry field (26 of Fig. 2). In particular, each entry (24 of Fig. 2) includes a work queue entry field (26 of Fig. 2), a first link field (28 of Fig. 2), and a second link field (30 of Fig. 2).

The first link field (28 of Fig. 2) stores a first entry identifier (e.g., "2" for Entry 0) referencing either a subsequently transmitted work queue entry (e.g., T="2" for Entry 0, "3" for Entry 2, "n" for Entry 3, and "4" for Entry "n") (page 10, lines 16-19), or the corresponding entry for the case of the last transmitted work queue entry (e.g., T= "4" for Entry 4) (page 10, lines 14-16). The first link identifiers (e.g., "2", "3", "n", "4", "4") in the respective first link fields (28 of Fig. 2) form a first linked list (32a of Fig. 2) specifying a transmit sequence of the transmitted work queue entries (page 10, lines 11-14 and 19-21).

The second link field (30 of Fig. 2) is configured for storing a second entry identifier (e.g., "n" for Entry 2) that references either the corresponding entry based on being the corresponding work queue entry receiving an acknowledgement (e.g., Ack = "4" for Entry 4 as the last received acknowledgement), or another entry having received a subsequent acknowledgement (e.g., Ack="2" for Entry 0, "n" for Entry 2, "4" for Entry "n") (page 10, line 23 to page 11, line 4). A second linked list (32b of Fig. 2) is generated that specifies an

acknowledgement sequence of the transmitted work queue entries (page 11, lines 1-8).

Hence, transmitted work queue entries awaiting receipt of acknowledgment messages can be efficiently managed and maintained using multiple fields (e.g., 28, 30) configured for specifying respective linked lists (e.g., 32a, 32b) for respective attributes of the work queue entries, optimizing memory utilization within the channel adapter (page 11, lines 5-8). Hence, a single table can be used to store multiple attributes of transmitted work queue entries, including the sequence of transmission, plus the sequence of received acknowledgements (page 3, lines 13-14).

Hence, independent claim 1 specifies method in a channel adapter (12 of Fig. 1, page 4, lines 13-17) configured for communications with a server network system. The method comprises first storing (step 102 of Fig. 3A), in a table (20 of Fig. 2, page 10, lines 6-11) configured for storing multiple entries, an entry (24 of Fig. 2) having a work queue entry field (26 of Fig. 2, page 10, lines 6-11) that specifies a transmitted work queue entry, the entry including at least first (28 of Fig. 2) and second link fields (30 of Fig. 2) each configured for referencing another entry in the table (page 10, lines 11-14 and 22-24), the first storing step including storing (102) in the first link field a first entry identifier (e.g., "2" for Entry 0, "3" for Entry 2, etc.) for one of the transmitted work queue entry (e.g., T=4 for Entry 4) and a subsequently transmitted work queue entry (e.g., "2" for Entry 0, "3" for Entry 2, "n" for Entry 3, and "4" for Entry "n") (page 10, lines 16-19) relative to the transmitted work queue entry (page 10, lines 11-21), wherein the first link entry identifiers in the respective first link fields form a first linked list (32a of Fig. 2) specifying a transmit sequence of the transmitted work queue entries (page 10, lines 11-14 and 19-21). The method further includes detecting (110 of Fig. 3B) an acknowledgement for at least a first of the transmitted work queue entries stored in the table (page 10, lines 24-28), and generating (112, 114, 116 of Fig. 3B, to page 10, line 22 to page 11, line 2) in the table a second linked list (32b) specifying an acknowledgement sequence of the transmitted work queue entries by second storing, in the second link field of the entry corresponding to the first transmitted work queue entry, a second entry identifier (e.g., "n" for Entry 2) based on the detecting step, the second entry identifier specifying one of the first

transmitted work queue entry (steps 112, 114, 118 of Fig. 3B) and an entry having received a subsequent acknowledgement relative to the detected acknowledgement (step 116 of Fig. 3B).

Claim 2 adds to the method of claim 1, wherein the first storing step includes storing the entry in a send queue table by a free buffer manager (76 of Figs. 1 and 2, page 9, lines 11-18 and page 10, lines 3-5).

Claim 3 adds to the method of claim 1, further comprising parsing (120 of Fig. 3B) the second linked list to determine transmitted work queue entries awaiting acknowledgement (page 11, lines 1-4).

Claim 4 adds to the method of claim 1, wherein the detecting step includes detecting the acknowledgement according to InfiniBand™ protocol (page 4, lines 14-17; acknowledgement detector 36 of Fig. 2 receives acknowledgements from InfiniBand™ network, page 10, lines 24-28).

Independent claim 5 specifies a channel adapter (12 of Fig. 1) comprising a table (20 of Fig. 2) configured for storing entries (24 of Fig. 2) identifying respective work queue entries having been transmitted according to a service protocol requiring receipt of an acknowledgment message (page 10, lines 6-8). Each entry includes: (1) a work queue entry field (26 of Fig. 2) configured for specifying the corresponding work queue entry (page 10, lines 8-11), (2) a first link field (28 of Fig. 2) configured for specifying a first entry identifier (e.g., "2" for Entry 0) referencing one of the corresponding entry (page 10, lines 14-16) and another entry for a subsequently transmitted work queue entry relative to the corresponding entry (e.g., "2" for Entry 0, "3" for Entry 2, "n" for Entry 3, and "4" for Entry "n") (page 10, lines 16-19), and (3) a second link field (30 of Fig. 2) configured for storing a second entry identifier (e.g., "n" for Entry 2) referencing one of the corresponding entry (e.g., "4" for Entry 4 as the last received acknowledgement) and another entry having received a subsequent acknowledgement (e.g., "2" for Entry 0, "n" for Entry 2, "4" for Entry "n") (page 10, line 23 to page 11, line 4). The channel adapter further includes an acknowledgement detector (36 of Fig. 2) configured for detecting the acknowledgements for the work queue entries (page 10, lines 24-28). The channel adapter also includes a table manager (22 of Fig. 2, page 10, line 28 to page 11, line 4) configured for adding

the table entries based on transmission of the respective work queue entries (102 of Fig. 3A), the table manager configured for inserting the corresponding first entry identifier based on the subsequently transmitted work queue entry (106 of Fig. 3A), the table manager configured for inserting the corresponding second entry identifier based on the entry having received the subsequent acknowledgement (112, 114, 116 of Fig. 3B), the first and second link fields forming first (32a) and second (32b) linked lists identifying a transmit sequence of the transmitted work queue entries and an acknowledgement sequence of the transmitted work queue entries, respectively (page 10, line 28 to page 11, line 4).

Claim 6 adds to the channel adapter of claim 5, wherein the acknowledgement detector is configured for detecting the acknowledgement according to InfiniBand™ protocol (page 4, lines 14-17; acknowledgement detector 36 of Fig. 2 receives acknowledgements from InfiniBand™ network, page 10, lines 24-28)..

Claim 7 adds to the apparatus of claim 5, wherein the table manager is configured for parsing the second linked list to determine transmitted work queue entries awaiting acknowledgement (120 of Fig. 3B, page 11, lines 1-4).

6. Grounds of Rejection to be Reviewed on Appeal:

A. Whether claims 1 and 3-7 are unpatentable under 35 U.S.C. §102(e) as having been anticipated in view of U.S. Patent No. 6,611,883 to Avery.

7. Arguments:

A. **Claims 1 and 5 are not anticipated Under 35 U.S.C. §102 in view of Avery**

In the Final Office Action, the Examiner rejected independent claims 1 and 5 under 35 USC §102(e) in view of Avery. Claims 1 and 5 are neither anticipated nor rendered obvious by Avery for the following reasons.

A1. **Avery Neither Discloses Nor Suggests The Claimed First Link Entry Identifiers of Respective Table Entries Forming a First Linked List**

**Specifying a Transmit Sequence of the Transmitted Work Queue Entries, or
Second Link Entry Identifiers of the Respective Table Entries Forming a
Second Link List Specifying an Acknowledgement Sequence of the
Transmitted Work Queue Entries**

Each of the independent claims 1 and 5 specify the operation of storing entries in a table, in a channel adapter: claim 1 specifies a method *in a channel adapter*, and claim 5 *a channel adapter comprising a table*.

Independent claims 1 and 5 also specify that *each table entry* includes: (1) a work queue entry field that specifies a transmitted work queue entry (WQE) (claim 5 specifies that the entries identify respective WQEs having been transmitted); (2) a first link field configured for specifying a first entry identifier for one of (i.e., either) the corresponding transmitted WQE and (i.e., or) a subsequently transmitted WQE relative to the corresponding entry; *and* (3) a second link field configured for specifying a second entry identifier for one of (i.e. either) the corresponding WQE and (i.e., or) a WQE having received a subsequent acknowledgement.

Independent claims 1 and 5 further specify that the first link fields form a first linked list (e.g., 32a of Fig. 2) *specifying a transmit sequence of the transmitted WQEs*. Hence, the claimed storage of first entry identifiers identifying *entries* of either the corresponding *transmitted WQE* or a *subsequently transmitted WQE* enables the formation of the first linked list specifying the transmit sequence of the transmitted WQEs.

Independent claims 1 and 5 further specify that the second link fields form a second linked list (e.g., 32b of Fig. 2) *specifying an acknowledgement sequence of the transmitted WQEs*. Hence, the claimed storage of second entry identifiers identifying *entries* of either the corresponding *transmitted WQE* or another entry *having received a subsequent acknowledgement* enables the formation of the second linked list specifying the acknowledgement sequence of the transmitted WQEs.

The Examiner has the burden of establishing that Avery discloses each and every element of the claim such that the identical invention must be shown in as complete detail as is contained

in the claim.¹ Further, Further, anticipation cannot be established based on a piecemeal application of the reference, where the Examiner picks and chooses isolated features of the reference in an attempt to synthesize the claimed invention.² In other words, it is not sufficient that a single prior art reference discloses each element that is claimed, but the reference also must disclose that the elements are arranged as in the claims under review. *In re Bond*, 15 USPQ2d 1566, 1567 (Fed. Cir. 1990) (citing *Lindemann Maschinenfabrik GmbH*).

In other words, the Examiner has the burden of establishing not only that Avery discloses a first linked list specifying a transmit sequence of the transmitted work queue entries and a second linked list specifying an acknowledgement sequence of the transmitted work queue entries, but also that each table entry includes the first and second link fields used for formation of the respective first and second linked lists. As described below, Avery discloses none of these claimed features.

A1.(a) Avery Does Not Disclose or Suggest the Claimed Entry Having A Work Queue Entry Field, a First Link Field, and a Second Link Field

The rejection fails to demonstrate that Avery discloses the claimed entry having a work queue entry (WQE) field, a first link field, and a second link field, as claimed. In particular, the rejection relies on the Infiniband address map 754 of Fig. 7 as disclosing the claimed “entry” that

¹As specified in MPEP §2131: “‘A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference’ *Verdegaal Bros. V. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). ... ‘The identical invention must be shown in as complete detail as is contained in the ... claim.’ *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).” MPEP 2131 (Rev. 3, Aug. 2005, at p. 2100-76).

² “Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim.” *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). “Anticipation cannot be predicated on teachings in the reference which are vague or based on conjecture.” *Studiengesellschaft Kohle mbH v. Dart Industries, Inc.*, 549 F. Supp. 716, 216 USPQ 381 (D. Del. 1982), *aff’d.*, 726 F.2d 724, 220 USPQ 841 (Fed. Cir. 1984).

includes a work queue entry field, and first and second link fields.

However, Avery describes element 754 in Figure 7 as an InfiniBand Address Map 754 that is used to associate Work Queue Entries (WQEs) with regions in the PCI address space. In particular, the Address Map of Figure 7 illustrates **two entries**, namely entries 756 and 758, that are used to associate work queue entries 703 and 705 with regions in the PCI address space 720:

The InfiniBand address map 754 is used to associate *work queue entries, such as entries 703 and 705* with regions in the PCI address space 720. The InfiniBand address map 754 is *a data structure* that is stored locally in the InfiniBand-PCI bridge 324 and *has a plurality of entries of which entries 756 and 758 are shown*. Each entry is associated with a region in the PCI address space 720 and holds the initial segment address for each region that is mapped into the system virtual memory address space, through the host channel adapter TPT.

(Col. 8, lines 55-64).

Avery recognizes that the “work queue entries” 703 and 705 are not table entries (in that they are not stored in the Address Map 754), but rather are InfiniBand™ data units that represent pending instructions that are to be performed via the InfiniBand™ network 300 of Fig. 3 (see, e.g., col. 2, lines 40-56 and col. 6, lines 10-30) which are placed in a “Send Work Queue” 311 of Fig. 3 or “Receive Work Queue” 312 of Fig. 3, depending on whether the transfer is a write request or a read request (col. 6, lines 10-48).

Avery describes that address map entry (e.g., 756, 758) in the Address Map 754 of Fig. 7 includes a queue pair pointer (QP PTR 766) that identifies a corresponding work queue entry (e.g., 702, 704): “[e]ach address map entry 756 also includes a pointer 766 to the work queue assigned to the address space mapping.” (Col. 9, lines 3-4). Hence, a particular work queue entry (e.g., 703, 705) is identified by its queue pair pointer (QP PTR) in the corresponding address map entry (e.g., 756, 758) (column 9, lines 3-7 and 46-47).

Although Avery describes that the address map entry (e.g., 756, 758) in the Address Map 754 of Fig. 7 includes a queue pair pointer (QP PTR 766) that identifies a queue pair (e.g., 702, 703), the address map entry (e.g., 756, 758) does not also include the claimed first link field in combination with the claimed second link field.

Rather, the address map entries (e.g., 756, 758) further consist of a valid flag field (F) 760, an error flag field (E) 762, and an “R-Key” field 764 that references the memory region used in specifying the limits of the virtual memory address space mapping and TPT address translation between the PCI address space and the system memory physical address (col. 8, lines 1-9 and 65 to col. 9, line 14).

Hence, none of the address map entries (e.g., 756, 758) of the Address Map 754 disclose the claimed first and second entry identifiers, and for this reason alone the §102 rejection should be reversed because Avery fails to disclose the claimed feature, *as arranged in the claim*, that each entry in the Address Map 754 includes, *in addition to the work queue entry field*, a first link field specifying a first entry identifier for a transmitted work queue entry, and a second link field second entry identifier specifying the corresponding entry or an entry having received a *subsequent acknowledgement*.

The rejection asserts on pages 3-4 of the Final Rejection that reference 766 in Fig. 7 of Avery is an example of “a pointer that references another entry in the table [sic] to work queue entry 703” and that “715 [is] a second pointer that references to a scoreboard that tracks acknowledgement sequence.” As apparent from the foregoing, however, the work queue entries 703 and 705 of Fig. 7 are not part of the entries 756 and 758 because the work queue entries 703 and 705 are referenced by the queue pair pointers 766. In other words, the Address Map 754, *as a distinct data structure* (see col. 8, lines 57-59) includes in each entry (e.g., 756, 758) a corresponding pointer 766 that “points to” *another data structure*, namely the associated work queue entry (e.g., 703 or 705) in the send queue 311 or the receive queue 312 of Fig. 3 (which form a “queue pair”) (col. 6, lines 10-30).

Further, each work queue entry (e.g., 703) includes a “DMA scoreboard pointer” 715 that references *another data structure*, namely a DMA context scoreboard 770 (col. 9, lines 58-62): the scoreboard 770 tracks the data tags associated with each data packet that is associated with the given work queue entry (col. 10, lines 54-59). Hence, the DMA context scoreboard 770 tracks the acknowledgments for a given work queue entry.

Avery consistently identifies the DMA context scoreboard 770 as a distinct data structure

(see, e.g., Abstract at lines 1-4; col. 3, lines 18-21; col. 9, line 58 to col. 10, line 6). As described in further detail below, the DMA context scoreboard 770 tracks the number of acknowledgements and not the claimed *acknowledgement sequence*.

Therefore, the Address Map 754, the send queue 311 and receive queue 310 that store the work queue entries (e.g., 703, 705), and the DMA context scoreboard 770 each are distinct data structures that reference each other by pointers 766 and 715. Consequently, Avery neither discloses nor suggests *the claimed entry* including the claimed work queue entry field, the claimed first link field, *and* the claimed second link field.

A1.(b) Avery Does Not Disclose or Suggest the Claimed First Linked List Specifying the Transmit Sequence, or the Claimed Second Linked List Specifying the Acknowledgement Sequence

As indicated above, the rejection asserts on pages 3-4 of the Final Rejection that reference 766 in Fig. 7 of Avery is an example of “a pointer that references another entry in the table [sic] to work queue entry 703” and that “715 [is] a second pointer that references to a scoreboard that tracks acknowledgement sequence.” The rejection also asserts on pages 3-4 that a first linked list is disclosed by “a first pointer referencing to an entry”.

Even though the Examiner is applying an unreasonable interpretation of the claimed terms “linked list” (discussed below), Avery still does not disclose or suggest a first linked list *specifying the transmit sequence*, or a second linked list *specifying the acknowledgement sequence*, as claimed.

Rather, Avery identifies the work queue entries “in the order that they were placed in the send work queue”, and not in the order they were *transmitted* (col. 6, lines 49-50). Moreover, as described above the pointer 766 provides no information regarding any transmit sequence, but merely associates an Address Map Entry with a Work Queue Entry (“[e]ach address map entry 756 also includes a pointer 766 to the work queue assigned to the address space mapping.” (Col. 9, lines 3-4)). Hence, the pointer 766 does not disclose or suggest a *transmit sequence*, as

claimed, regardless of whether the single pointer could be considered a “linked list”, as asserted by the Examiner.

In addition, the rejection erroneously assumes that the pointer 715 to the DMA context scoreboard constitutes a “second linked list specifying an acknowledgement sequence”.

However, Avery explicitly specifies that the DMA scoreboard 770 keeps tracks of outstanding DMA requests based on the size of the data transfer or the number of prefetches.

Specifically, Avery specifies that the DMA scoreboard 770 “is a data structure that holds the DMA context and tracks outstanding DMA requests to insure that all outstanding requests are completed. The DMA scoreboard 770 contains data tags 774 to track data returning from outstanding DMA read requests and ACK flags which track whether DMA writes have been completed.” (Col. 9, lines 59-65).

Further, the DMA scoreboard 770 includes a “prefetch section that includes the number of prefetches 776 and a size section 775 (col. 9, line 67 to col. 10, line 1). Avery describes prefetching with reference to Figs. 7 and 9B by increasing a prefetching address by the prefetching size 774 until the DMA transfer size 775 is equal to the work queue entry size request 710:

Since these [DMA read] requests are acted on asynchronously by the InfiniBandSM bus system, they must be coordinated by the DMA context scoreboard 770 *which stores the number and size of additional read requests that must be made in fields 774 and 775, respectively*.

(Col. 10, line 65 to col. 11, line 3)

A tag that uniquely identifies the request is stored in field 774. ... The data transfer software then monitors, in step 922, the returning data packets until a response packet corresponding to the outstanding request is received. The response packet can be examined to determine whether it corresponds to the initial read request.

A prefetch is performed using the same work queue pair that was used to perform the initial DMA read. When a response packet is received from the initial DMA read request, the process then proceeds to step 924 where the DMA scoreboard is updated by adding the prefetch size stored in field 774 to the current address and storing the result in next address field 772. *A check is then made in step 926 to determine whether the DMA*

transfer size is equal to the work queue entry size request 710. If the sizes are equal, the process finishes in step 928.

If prefetch data remains to be retrieved, the next prefetch request is initiated in step 930 by combining the next address in DMA scoreboard field 772 with the R-Key 764 in the address map 754 as schematically indicated by block 760 to produce a new virtual address which is pushed into work queue entry 703. The process then returns to step 920 in which the new prefetch read request is generated. *Operation continues in this manner until the size 710 in the work queue entry 703 is exhausted as determined in step 926* or, alternatively, until internal buffer resources are consumed.

(Col. 11, lines 7-36).

Hence, Avery specifies that, in response to an acknowledgement that matches the acknowledgement tag 776, the next address 770 is incremented by the prefetch size 774, and a new fetch address is created at block 760 based on combining the next address 770 with the R-Key 708, until the DMA transfer size 775 is equal to the work queue entry size request 710.

Hence, Avery provides no disclosure or suggestion of storing an *acknowledgement sequence* of transmitted work queue entries, as claimed; rather, Avery simply determines whether an acknowledgement for a specific data packet has been received, and continues to output additional packets until the DMA transfer size 775 equals the work queue entry size request 710.

Hence, there is no disclosure or suggestion of the claimed first linked list *specifying a transmit sequence* of the transmitted work queue entries generated from first link fields, in combination with a second linked list *specifying an acknowledgement sequence of the transmitted work queue entries* generated from second link fields, as claimed.

Anticipation cannot be established based on a piecemeal application of the reference, where the Examiner picks and chooses isolated features of the reference in an attempt to synthesize the claimed invention. “Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim.” *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). Hence, it is not sufficient that a single prior art reference discloses each

element that is claimed, but the reference also must disclose that the elements are arranged as in the claims under review. *In re Bond*, 15 USPQ2d 1566, 1567 (Fed. Cir. 1990) (citing *Lindemann Maschinenfabrik GmbH*).

For these and other reasons, the §102 rejection of independent claims 1 and 5 should be withdrawn.

B. The Examiner's Interpretation of the Claimed "Linked List" Is Unreasonable Because It Disregards Claim Terms and is Inconsistent With the Specification and the Interpretation by Those Skilled In The Art

Although the Examiner is permitted to interpret the claims broadly, that interpretation must be reasonable. The broadest *reasonable* interpretation must be (1) consistent with the specification, and (2) consistent with the interpretation that those skilled in the art would reach.³ Hence, the claimed "first linked list" must be derived from the first link fields in the table, and the "second linked list" must be derived from the second link fields in the same table.

However, the Examiner is deliberately disregarding the specification and the established interpretation that those skilled in the art would reach to reduce the claimed "linked list" to a simple "pointer". As demonstrated below, this interpretation not only ignores the specification and the established interpretation by those skilled in the art, but it also disregards explicit claim language.

The Examiner provides a tortured and unreasonable interpretation of the claimed "first linked list" and "second linked list" by ignoring the very evidence he submitted to support his

³"During patent examination, the pending claims must be 'given their broadest reasonable interpretation consistent with the specification.'" MPEP §2111 at 2100-46 (Rev. 3, Aug. 2005) (*quoting In re Hyatt*, 211 F.3d 1367, 1372, 54 USPQ2d 1664, 1667 (Fed. Cir. 2000)).

"The broadest reasonable interpretation of the claims must also be consistent with the interpretation that those skilled in the art would reach." MPEP §2111.01 at 2100-47 (Rev. 3, Aug. 2005) (*citing In re Cortright*, 165 F.3d 1353, 1359, 49 USPQ2d 1464, 1468 (Fed. Cir. 1999)).

assertion.

In particular, the Examiner proposed a definition page 2 of the October 26, 2005 Final Action by citing the Microsoft Computer Dictionary: according to the Examiner, the Microsoft Computer Dictionary defines a linked list as “a list of nodes or *elements of a data structure connected by pointers*.” Hence, the definition relied on by the Examiner requires a linked list to include: (1) a list of *elements of a data structure*; (2) the elements of the data structure are *connected*; and (3) the elements of the data structure are *connected by pointers*.

U.S. Patent No. 6,697,380 (of record in this application) also illustrates in Fig. 3 a first linked list of elements 56 of *a data structure 27* that are linked by first pointers 62, and a second linked list of the elements 56 of the data structure 27 linked by second pointers 64.

The Attached Exhibit A (originally submitted in Applicant’s December 27, 2005 Response After Final) (page 1 of a definition of “Linked list” at http://en.wikipedia.org/wiki/Linked_list#Doubly-linked_lists by Wikipedia) also describe linked lists also require linking to the same data type: “A linked list is a self-referential datatype because it contains a pointer or link to another data of the same type.” The Microsoft and Wikipedia definitions both are consistent in their requirement that the same data type is referenced, since storage of elements in the same data structure implies the elements are of the same type.

The specification also illustrates in Figure 2 that the elements of the linked list 32a and 32b are (1) *within the same data structure 20*; (2) connected, and (3) connected by pointers from first link fields 28 that identify subsequently transmitted work entries (for linked list 32a) or pointers from second link fields 30 identifying subsequent acknowledgements (for linked list 32b).

Finally, the claims explicitly specify: the first linked list is formed by the *first entry identifiers in the respective first link fields of the entries of the table*, and the second linked list is formed by the *second entry identifiers in the respective second link fields of the entries of the table*. Hence, the claims explicitly specify that first linked list is formed from the same data type (i.e., the first link fields) within the same data structure (i.e., the table), and the second linked list is formed from the same data type (i.e., the second link fields) within the same data structure

(i.e., the table).

The Examiner's interpretation deliberately ignores the above-described requirements in a linked list that (1) the elements must be within *the same data structure*, and that (2) the pointer must point to another data *of the same type*. For example, the Examiner ignores the claim language in the Advisory Action by asserting that the essential requirement of linked lists (linking to another element in the same data structure and of the same type of element) is not claimed.

To the contrary, these elements are claimed because all the linked elements are *in the claimed table*, and because the first linked list is formed from identifiers in the *first link field* (identifying transmitted work queue entries), and the second linked list is formed from identifiers in the *second link field* (identifying received acknowledgements).

Further, the Examiner deliberately ignores the above-described features of Avery that: (1) the Address Map 754, the WQEs 703/705, and the DMA context scoreboard 770 are *distinct data structures*; and (2) the Queue Pair Pointer 766 (let alone the entries 756/758 of Address Map 754) is not of the same data type as the WQE 703/705, and the DMA Scoreboard Pointer 715 (let alone the WQE 703/705) is not of the same data type as the DMA context scoreboard 770.

Hence, the Examiner's assertion on pages 3-4 of the Final Action is fundamentally flawed because the Examiner simply asserts that "the process therefore forms a linked list simply by referencing another entry by a pointer", while ignoring the essential teaching that the pointers should link to another element in the same data structure and be of the same type of element; rather, the pointers relied on by the Examiner are pointers between different data structure that have *different data types*.

The Examiner's interpretation of the claimed linked lists is inconsistent with the explicit claim language, and therefore is per se improper because "[a]ll words in a claim must be considered in judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970).

Further, the Examiner has failed to provide any evidence whatsoever to support his

interpretation of “linked list” as “simply by referencing another entry by a pointer”. In fact, the Examiner’s interpretation is inconsistent with the interpretation by Microsoft Computer Dictionary propounded by the Examiner, the art of record (USP 6,697,380), and the known interpretations in the art as demonstrated by the attached Exhibit A. Further the Examiner’s interpretation is inconsistent with the specification, which illustrates the linked lists 32a referencing the same data types within the same data structure.

The Examiner’s interpretation is therefore unreasonable because it ignores essential claim limitations, is inconsistent with the specification, and inconsistent with the interpretation those skilled in the art would reach: “the PTO’s interpretation of claim terms should not be so broad that it conflicts with the meaning given to identical terms in other patents from analogous art.” *In re Cortright*, 49 USPQ2d 1464, 1467 (Fed. Cir. 1999).

For these and other reasons, the §102 rejection of independent claims 1 and 5 should be reversed. The dependent claims are allowable in view of their dependency from claims 1 or 5.


Conclusion

For the reasons set forth above, it is clear that Appellant’s claims 1-7 are patentable over the reference applied. Accordingly the appealed claims 1-7 should be deemed patentable over the applied reference. It is respectfully requested that this appeal be granted and that the Examiner’s rejections be reversed.

To the extent necessary, Appellant petitions for an extension of time under 37 C.F.R. 1.136 and 37 C.F.R. 41.37(e). Please charge any shortage in fees due in connection with the filing of this paper, including any missing or insufficient fees under 37 C.F.R. 1.17(a) or 41.20(b)(2), to Deposit Account No. 50-0687, under Order No. 95-507, and please credit any excess fees to such deposit account.

Respectfully submitted,

Manelli Denison & Selter, PLLC

A handwritten signature in black ink, appearing to read 'L. R. Turkevich', with a stylized flourish at the end.

Leon R. Turkevich
Registration No. 34,035

Customer No. 20736

CLAIMS APPENDIX

1. (ORIGINAL) A method in a channel adapter configured for communications with a server network system, the method comprising:

first storing, in a table configured for storing multiple entries, an entry having a work queue entry field that specifies a transmitted work queue entry, the entry including at least first and second link fields each configured for referencing another entry in the table, the first storing step including storing in the first link field a first entry identifier for one of the transmitted work queue entry and a subsequently transmitted work queue entry relative to the transmitted work queue entry, wherein the first link entry identifiers in the respective first link fields form a first linked list specifying a transmit sequence of the transmitted work queue entries;

detecting an acknowledgement for at least a first of the transmitted work queue entries stored in the table; and

generating in the table a second linked list specifying an acknowledgement sequence of the transmitted work queue entries by second storing, in the second link field of the entry corresponding to the first transmitted work queue entry, a second entry identifier based on the detecting step, the second entry identifier specifying one of the first transmitted work queue entry and an entry having received a subsequent acknowledgement relative to the detected acknowledgement.

2. (ORIGINAL) The method of claim 1, wherein the first storing step includes storing the entry in a send queue table by a free buffer manager.

3. (ORIGINAL) The method of claim 1, further comprising parsing the second linked list to determine transmitted work queue entries awaiting acknowledgement.

4. (ORIGINAL) The method of claim 1, wherein the detecting step includes detecting the acknowledgement according to InfiniBand™ protocol.

5. (ORIGINAL) A channel adapter comprising:
- a table configured for storing entries identifying respective work queue entries having been transmitted according to a service protocol requiring receipt of an acknowledgment message, each entry including:
- (1) a work queue entry field configured for specifying the corresponding work queue entry,
 - (2) a first link field configured for specifying a first entry identifier referencing one of the corresponding entry and another entry for a subsequently transmitted work queue entry relative to the corresponding entry, and
 - (3) a second link field configured for storing a second entry identifier referencing one of the corresponding entry and another entry having received a subsequent acknowledgement;
- an acknowledgement detector configured for detecting the acknowledgements for the work queue entries; and
- a table manager configured for adding the table entries based on transmission of the respective work queue entries, the table manager configured for inserting the corresponding first entry identifier based on the subsequently transmitted work queue entry, the table manager configured for inserting the corresponding second entry identifier based on the entry having received the subsequent acknowledgement, the first and second link fields forming first and second linked lists identifying a transmit sequence of the transmitted work queue entries and an acknowledgement sequence of the transmitted work queue entries, respectively.
6. (ORIGINAL) The channel adapter of claim 5, wherein the acknowledgement detector is configured for detecting the acknowledgement according to InfiniBand™ protocol.
7. (ORIGINAL) The apparatus of claim 5, wherein the table manager is configured for parsing the second linked list to determine transmitted work queue entries awaiting acknowledgement.

EVIDENCE APPENDIX:

Attached: Exhibit A (1 sheet) of Response After Final filed December 27, 2005

Linked list

From Wikipedia, the free encyclopedia.

In computer science, a **linked list** is one of the fundamental data structures used in computer programming. It consists of a sequence of nodes, each containing arbitrary data fields and one or two references ("links") pointing to the next and/or previous nodes. A linked list is a self-referential datatype because it contains a pointer or link to another data of the same type. Linked lists permit insertion and removal of nodes at any point in the list in constant time, but do not allow random access. Several different types of linked list exist: singly-linked lists, doubly-linked lists, and circularly-linked lists.

Linked lists can be implemented in most languages. Languages such as Lisp and Scheme have the data structure built in, along with operations to access the linked list. Procedural languages such as C, C++, and Java typically rely on mutable references to create linked lists.

Contents

- 1 History
- 2 Variants
 - 2.1 Linearly-linked List
 - 2.1.1 Singly-linked list
 - 2.1.2 Doubly-linked list
 - 2.2 Circularly-linked list
 - 2.2.1 Singly-circularly-linked list
 - 2.2.2 Doubly-circularly-linked list
 - 2.3 Sentinel nodes
- 3 Applications of linked lists
- 4 Tradeoffs
 - 4.1 Linked lists vs. arrays
 - 4.2 Doubly-linked vs. singly-linked
 - 4.3 Circularly-linked vs. linearly-linked
 - 4.4 Sentinel nodes
- 5 Linked list operations
 - 5.1 Linearly-linked lists
 - 5.1.1 Singly-linked lists
 - 5.1.2 Doubly-linked lists
 - 5.2 Circularly-linked lists
 - 5.2.1 Doubly-circularly-linked lists
- 6 Linked lists using arrays of nodes
- 7 Language support
- 8 Internal and external storage
 - 8.1 Example
- 9 Speeding up search
- 10 Related data structures
- 11 References
- 12 External links